Team 5

Contents

Introduction

Graph plotting is a very frequently used service, whether for students' use in classes or for business purposes to efficiently visualize large volumes of data, to be interpreted in multiple ways. Existing solutions like TeraPlot and MATLAB, while being powerful tools, are difficult to use and too expensive in many cases. Online plotters like Desmos, Mathway etc. are dependent on internet connections and put other restrictions on their use.

Graphi is a widget for function plotting and data visualization which provides inexpensive and powerful 2D/3D Graphing capabilities. It is great at quick visualization of mathematical functions while being entirely open source. It is a one-of-its-kind standalone graph plotter for Linux (can also be compiled cross-platform) which brings to users simple and efficient plotting.

Using ***Graphi***, the user can:

- Create multiple plots on the same graphs
- Re-plot the graphs using different parameters
- Have different features (like colour, line types, line width) for different function plots. These can be the default settings or user-specified
- View the graphs, and if not satisfied, re-plot the graph with different parameters
- Specify orientation of the axis for 3D plots
- Store the graph in file formats of your choice
- Export graphs as images to a file or the clipboard
- Specify graph axis labelling
- Scatter plotting in 2D and 3D
- Zoom, pan, and animate 3D graphs
- Create Cartesian or parametric 2D plots with various line style options
- View the graph in full screen mode
- Obtain the value of the function at a point

The following sections describe how to use the various ***Graphi*** features.

Overview

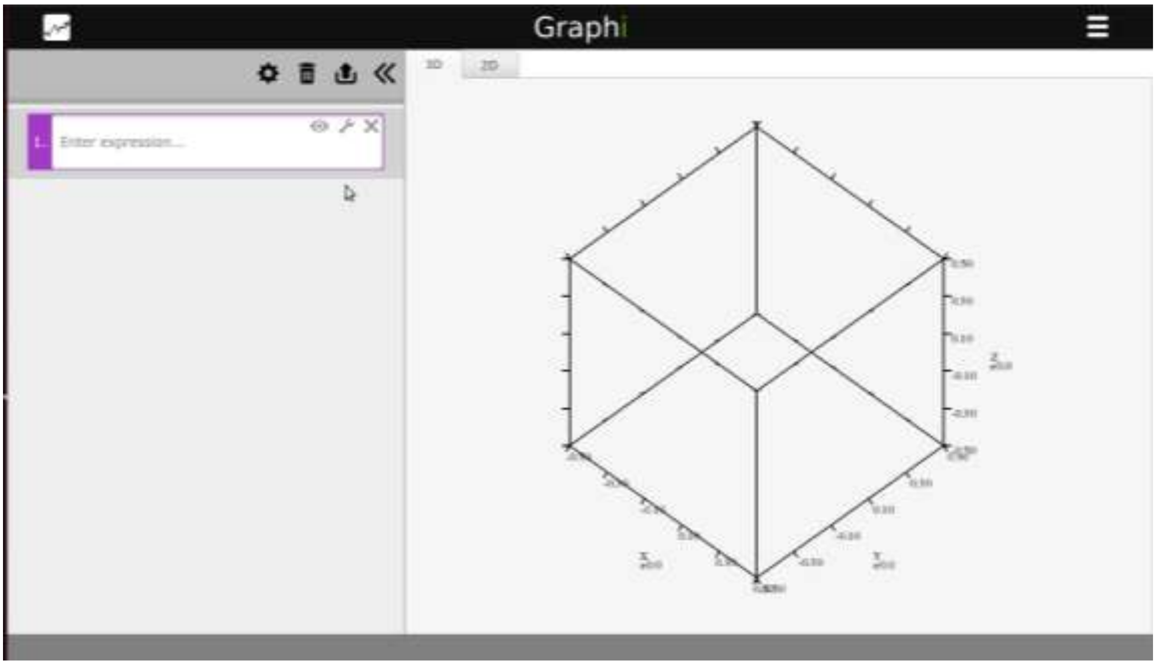


Fig. 1 Home Screen

Graphi is built on a stand-alone User Interface that allows for input of multiple functions for plotting on the same graph. It also allows for easy editing and deleting of various graph properties like graph colour, line width, format etc.

A ***Graphi*** project consists of a number of graphs that can be plotted either on the same axes or as sub-plots on multiple graphs. These options are all user specified and the UI is highly intuitive and interactive.

All options are accessed via the equation-box panel and the top app bar. Project creation, loading, and saving are handled via menu items on the File menu.

An inbuilt parser automatically evaluates the type of function that must be plotted (2D or 3D) and the function is plotted on a graph when the user presses the 'Enter' key in the text box.

1. Graphs & Plots

After creating a graph it can be modified in various ways, using the menu items from the Graph and Axes/Grid menus.

When there is more than one graph on a page, tapping anywhere within that graph makes it the selected graph, and any modifications performed will apply to that graph only. Typically, these modifications are performed via a feature-related popup found in a menu item on the settings bar. For example, graph plots are edited via the dropdown menu from the Settings (wrench) icon in the function text box.

2D Graphs contain Line Plots, and 3D graphs contain Surface Plots. In each case, plots can be either from an Analytical function or Tabular data. Analytical plots are defined via a script containing one or more lines of mathematical expressions. Tabular plots are defined via a table of data, which can be entered manually, pasted from the clipboard, or loaded from a text file. Once created, various plot features can be modified, e.g. line colour and style for line plots, surface colour, colour map, and texture overlay for surface plots. Besides plots, various other graph features can be modified. These include the colour of all graph components such as graph foreground, axis labels, titles and axis ranges.

A graph holder represents the area occupied by an uninitialized graph, and contains two tabs, labelled 2D Graph and 3D Graph, as shown in figure above.

2. User Interface

Graphi's user interface is minimalistic yet intuitive making it easy to use and understand. It consists of a sliding panel for the function text-boxes and the graph windows for plotting. Simple arrows on the panel allow for quick resizing and hiding of the panel.

Each text box has a specific border colour that matches the colour of the graph plotted. The function is typed into the text box and the user presses "Enter" to obtain a plot. The automatic parser allows conventional function scripting (E.g. 3sin(x) is interpreted as 3*sin(x)) making it easier to use.

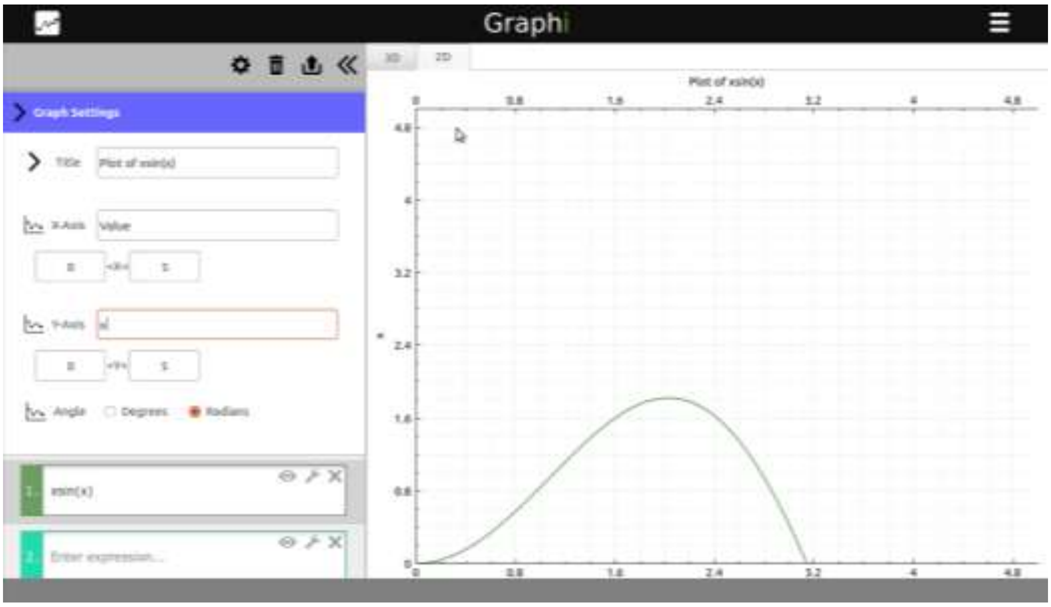
The graph plotted can be zoomed in or out using buttons and the scroll bar as an added functionality. Scrolling up or down will zoom in or out respectively on the graph. Panning can be achieved simply by pressing the scroll button on the mouse and dragging.

Clicking using the Left Mouse Button and dragging will pan the graph in 2D and rotate the same in 3D as shown.

The user interface supports touch enabled screens which is an emerging feature on many laptops and desktop PCs which makes the experience of the user much more interactive. The architecture of the software is such that the entire app can be easily extended for mobiles and tablet PCs.

3. Labelling

X, Y and Z axes have markings given by the mantissa of the value at the point. The exponent for each axis is specified in the legend of the graph. The axis labelling is also dynamic and can be changed using the graph settings option as shown below.



Title Plot of xsin(x)

X-Axis Value

Y-Axis a

Angle ○ Degrees ● Radians

1. xsin(x)

2. Enter expression...

Fig. Graph settings and axis labelling

4. 2D graphs

2D graphs are specified by two co-ordinate axes X and Y. After initializing a 2D graph, the next step is to add one or more line plots to it. This is done when the user enters a 2D function in the text box. The function plot is automatically created on pressing the "Enter" key and a window containing the plot appears on the screen as shown. When a tabular plot is added or selected, a two column grid of cells is provided for data entry.

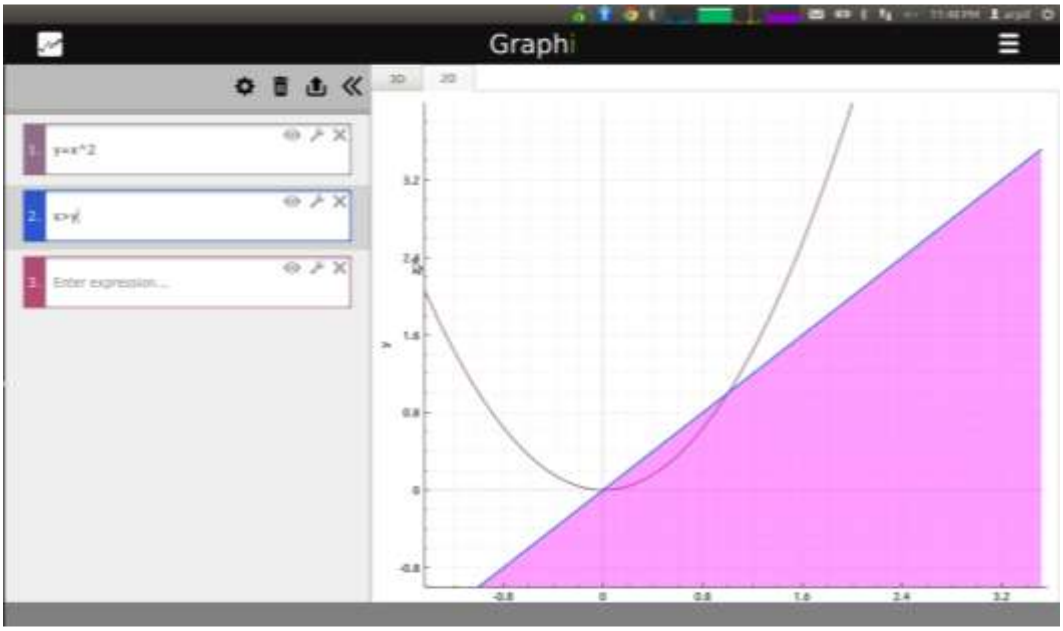
Line plots in a 2D graph can be defined in 3 coordinate forms for ***GRAPHI***. These are:

1. Cartesian, y = f(x). This is the normal x, y coordinate system.

2. Implicit, f(x,y) = k where k is a constant.

3. Parametric, $x = f_1(t)$, $y = f_2(t)$. This is the parametric definition of a line where both x and y are specified as functions of a parameter t.

4. Scatter plots: Data can be entered manually or from a file (.csv or .txt) to obtain a scatter plot of the points.



1. y=x^2

2. x>y

3. Enter expression...

Fig. 2D Plots of explicit and conditional expressions

5. 3D Graphs

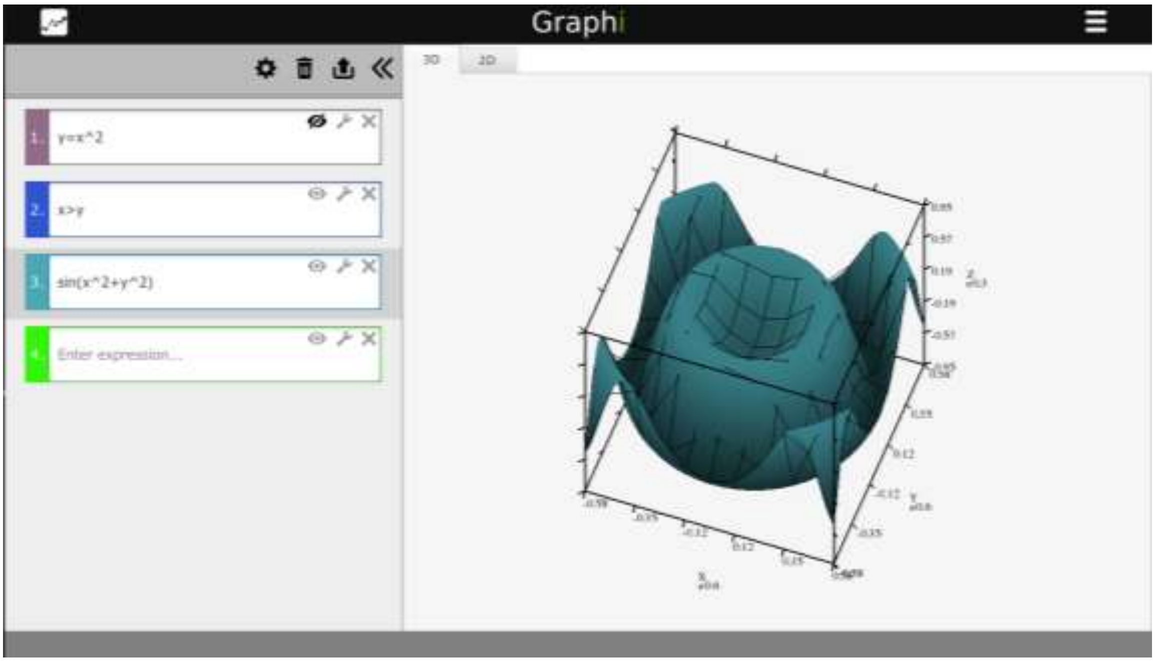
3D graph refers to one which has three axes: x, y and z. When an analytical plot is added or selected, a text box is made available for script entry. When a tabular plot is added or selected, a multi column grid of cells is provided for data entry. Pressing the "Enter" key will automatically plot the graph in the panel created on the left.

Surface plots in a 3D graph can be defined in normal cartesian coordinate form. ***Graphi*** supports only explicit plots for 3D graphs i.e.

$$\boldsymbol{z = f(x, y)}$$

$$\boldsymbol{z = sin(x^2 + y^2)} \textbf{ (Plot shown below)}$$

Scatter plots: Data can be entered manually or from a file (.csv or .txt) to obtain a scatter plot of the points.



1. y=x^2
2. x>y
3. sin(x^2+y^2)
4. Enter expression...

Fig. plot of sin(x^2 + y^2)

Note that plotting of implicit functions is NOT SUPPORTED for 3D functions.

5. Graph Settings

Graph settings are changed via dynamic menus associated with the wrench icon in the graph input text-box. Clicking on it gives three menus which give:

- Graph Settings: Here we can specify the range of each axes on the graph, give it a title and specify the step size for calculation. A smaller step size gives a more accurate plot of the graph but works slower in that case. A radio button allows the angles in the graph to be specified in either degrees or radians.
- Line settings: Settings associated with the line that plots the particular function on the graph (Line style, line width and line colour)
- Marker settings: Gives characteristics of the marker on the graph (Marker style, marker width and marker colour)
- Brush settings: Brush tool is used to provide shading to plot areas of the graph. We can specify the style, width and colour of the brush used to perform these operations from the brush settings tab.
- Zooming can be done by scrolling mouse wheel up and down
    - By default, X and Y axes are zoomed
    - Holding control zooms the Z axis
    - Holding shift and control zooms the Y axis
    - Holding alt and control zooms the X axis

- Hold down the middle mouse and move the mouse button to pan the plot
    - Horizontal motion changes the X range
    - Vertical motion changes the Y range
    - Holding down control changes the Z range
- Plot orientation is changed by holding down the mouse left button
    - Horizontal motion changes orbital angle
    - Vertical motion changes the elevation
- Value at a point
    - Values of a point on the plot can be determined by placing the pointer at the desired point and pressing control

6. Saving

The graphs plotted can be saved in the following formats:

- 2D:
    - Vector format .pdf
    - Raster format .jpeg, .png, .bmp
- 3D:
    - Vector format svg
    - Raster bmp

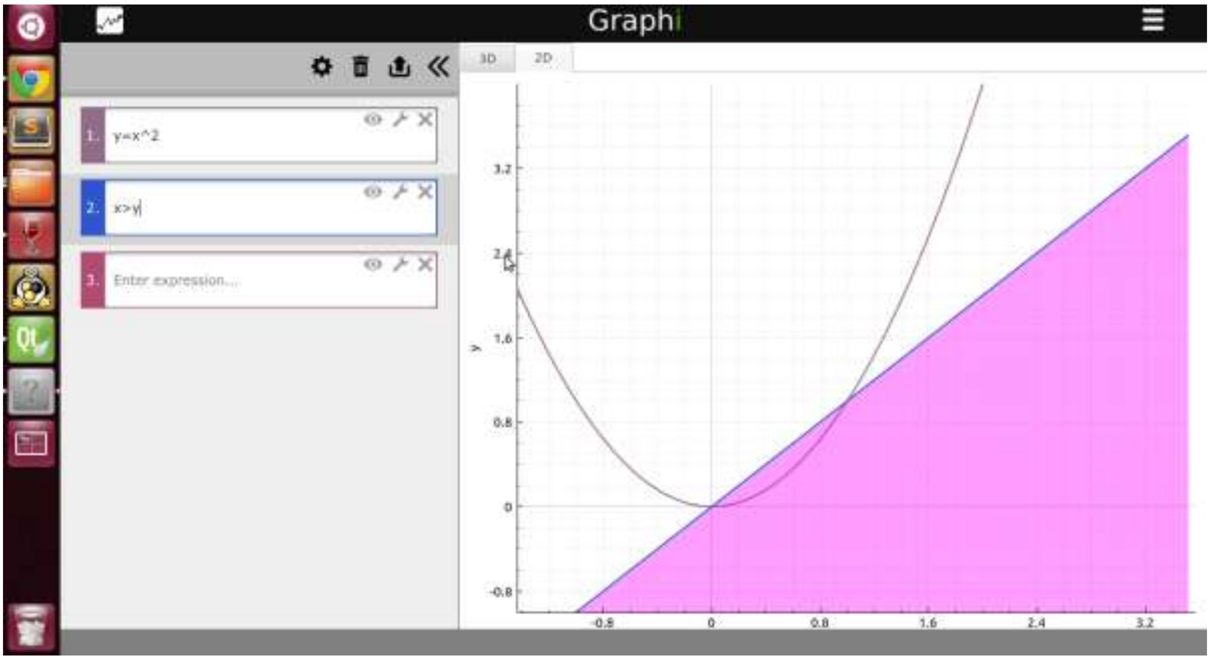
7. Scripting

Plotting of graphs involves creating one or more lines of text which assign a value to the dependent variable(s) for that plot type, usually as a function of the independent variable(s). Scripts can consist of more than one line (parametric) where the parser automatically evaluates the type of function (2D or 3D) thereby allotting the function entered to the next variable. When the “Enter” button is pressed, ***Graphi*** then evaluates the script for every value of the independent variable(s) and draws the plot. Taking examples:

$$\boldsymbol{y = x^2}$$

$$\boldsymbol{sin(x^2 + y^2)}$$

In the first expression, the parser interprets the above as a function of x only and the app automatically evaluates the function to be 2D, reading it as f(x) = y and plotting on the X-Y plane as shown.



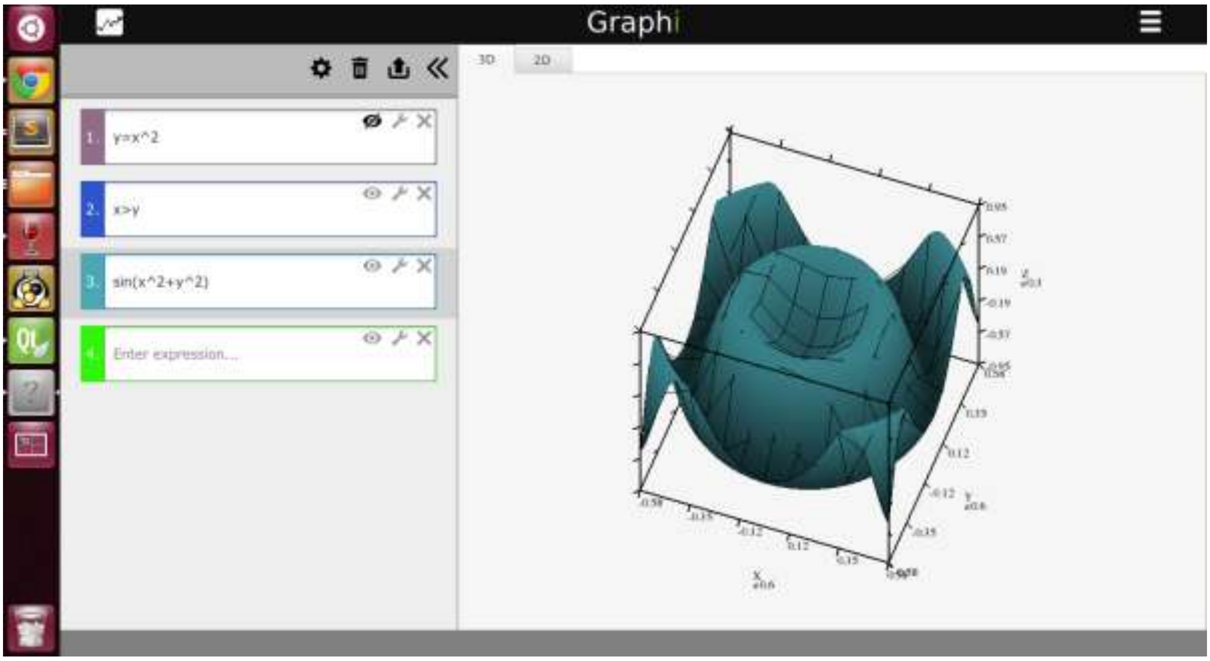
1. y=x^2

2. x>y

3. Enter expression...

Figure 1 Plot of y = x^2

The second expression shows a 3D surface plot of the function f(x, y) = sin(x^2 + y^2) = z as shown.



1. y=x^2

2. x>y

3. sin(x^2+y^2)

4. Enter expression...

Figure 2 Plot of sin(x^2 + y^2)

7.1 Arithmetic Syntax

When writing math expressions on paper, multiplication is often implied, as is the application of a function to a variable. An example might be:

$$y = 3\sin(x)$$

Or

$$y = 3 * sin(x)$$

If the first expression was entered as is, the ***Graphi*** script parser would still be able to correctly compute the function as given in expression 2.This allows for convenient readability by new users.
The use of the other arithmetic symbols +, -, /, requires no special considerations

7.2 Reserved Symbols

There are certain constants which are defined in ***Graphi*** which cannot be changed and can be used simply by writing the variable names.
e = 2.7182
pi = 3.1415

$\boldsymbol{3pi + sin(x)}$ **Will result in**
3*3.1415 + sin(x)

7.3 Function definitions for implicit, conditional and parametric plots

Graphi supports implicit plotting of functions in 2D. Functions can be input in the text box as follows.

$$x^3/6 + y^2/10 = 1$$

Conditional statements can also be implemented in the plotter by typing the statement as follows:

$$x > y$$

This gives a 2D plot where the region specified by the condition is shaded.

Parametric plots can be made in 2D by defining x and y as functions of "t ". Note that the only parametric variable allowed is "t". The range of "t" can also be specified after the expression in curly braces as shown below.

$$x = t, y = t^2, \{1, 7.8\}$$

$$x = t^2, y = t^3, z = sin(t^{0.4}), \{1, 2.556\}$$

Note here how the range is input to the program and how it compares to the range specification in normal functions.

All functions are comma separated. Use of spaces will result in an error.

7.4 Dealing with Singularities and Undefined Points

Graphi can plot functions with singularities and undefined points in the following manner:

- Points where the function goes to infinity:

$$y = \tan(x) \text{ Where x = } \frac{\pi}{2}$$

The function is plotted in the vicinity of the point and zooming in on the singularity will show the graph moving towards infinity.

- Points where the function is not defined for the point (Not in domain):

$$y = \frac{\sin(x)}{x} \; where \; x = 0$$

The function is plotted in the vicinity of the point and the curve is extrapolated from the points in the neighbourhood of the singularity to give a fitting curve i.e. the point outside the domain is ignored.

List of functions defined:

Function	**Description**	**Example**
sin()	Sine	sin(x)
cos()	Cosine	cos(x)
tan()	Tangent	tan(x)
cot()	Cotangent	cot(x)
sec()	Secant	sec(x)
cosec()	Cosecant	**cosec(x)**
sinh()	Hyperbolic sine	**sinh(x)**
cosh()	Hyperbolic cosine	**cosh(x)**
tanh()	Hyperbolic tangent	**tanh(x)**
ln()	Natural logarithm	**ln(x)**
lg()	Logarithm to base 2	**lg(x)**

log()	Logarithm to base 10	**log(x)**
sqrt()	Square root	**sqrt(x)**
exp()	Exponent	**exp(x)**
abs()	Absolute value	**abs(x)**

List of Operators Supported:

Operator	Description	Example
+	Sum	a+b
-	Difference	a-b
*	Product	a*b
/	Quotient	a/b
^	Power (a^b)	a^b
<	Greater than	a>b
>	Less than	a<b
>=	Greater than or equals	a>=b
<=	Less than or equals	a<=b